

**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A system for controlling an application process comprising:
  - an injector stored on a first computing device executing the application process;
  - redirect code placed by the injector in a memory of the first computing device ~~used by an application process, the redirect code allows access to secured data at a remote computer system by authorizing access with security in place at the remote computing system;~~ and
  - a library of redirect functions operable to be referenced by the redirect code during execution of the application process execution, wherein the redirect code is operable to (i) intercept a set of target at least one function calls call made by the application process to access secured data at a remote computer system, and (ii) execute at least one of the redirect functions for in place of the at least one intercepted target-function calls call so as to enable the application process, executing at the first computing device, to access the secured data at the remote computer system.
2. (Previously Presented) The system, as set forth in claim 1, wherein the injector is pushed to the first computing device executing the application process.
3. (Currently Amended) The system, as set forth in claim 1, wherein the ~~set of target at least one function calls call~~ comprises a socket function calls call.
4. (Original) The system, as set forth in claim 1, wherein the library of redirect functions comprises a dynamic link library.
5. (Previously Presented) The system, as set forth in claim 1, further comprising:
  - a secure environment having a plurality of resources;
  - a firewall securing all access to the plurality of resources in the secure environment; and
  - an access policy pushed to the first computing device executing the application process, the access policy identifying the resources authorized for access by the first computing device.

6. (Original) The system, as set forth in claim 5, wherein the application process comprises an application operable to communicate with the secure environment resources using an Internet transport protocol, the redirect code, and the redirect functions.

7. (Original) The system, as set forth in claim 1, wherein the application process comprises an email application.

8. (Original) The system, as set forth in claim 1, wherein the application process comprises a web browser application.

9. (Original) The system, as set forth in claim 1, wherein the application process comprises a file transfer application.

10. (Currently Amended) A method for controlling an application process comprising:

pushing an injector to a first computing device enabled to execute ~~executing the~~ application process;

starting an execution of the application process;

interrupting the execution of the application process;

injecting, via the injector, a redirect code into the application process, the redirect code allows access to secured data at a remote computer system by authorizing access with security in place at the remote computing system;

executing the redirect code in the application process to reference a redirect library of redirect functions[[]] so that upon resuming the execution of the application process, the redirect code is operable to (i) intercept, and intercepting at least one target function calls call made by the application process to access secured data at a remote computer system, and (ii) execute ~~executing~~ at least one redirect function in place of the at least one target function calls call so as to enable the application process, executing on the first computing device, to access the secured data at the remote computer system.

11. (Currently Amended) The method, as set forth in claim 10, wherein injecting ~~a the~~ redirect code further comprises: ~~starting the application process; interrupting the execution of the application process; and~~ injecting the redirect code into a memory space of the application process.

12. (Currently Amended) The method, as set forth in claim 10, wherein ~~injecting a redirect code further starting and interrupting the execution of the application process~~ comprises[[:]] starting the execution of the application process using a debug option[[:]], and catching an exception thrown by the application process; and wherein injecting the redirect code comprises locating memory space in the application process[[:]], injecting the redirect code into the memory space of the application process[[:]], and ~~set~~ setting an instruction pointer to the redirect code.

13. (Currently Amended) The method, as set forth in claim 10, wherein ~~injecting a redirect code further starting and interrupting the execution of the application process~~ comprises[[:]] starting the execution of the application process using a suspend option; and wherein injecting the redirect code comprises creating memory space in the application process[[:]], injecting the redirect code into the memory space of the application process[[:]], and ~~set~~ setting an instruction pointer to the redirect code.

14. (Currently Amended) The method, as set forth in claim 10, wherein ~~injecting a redirect code further starting and interrupting the execution of the application process~~ comprises[[:]] starting the execution of the application process using a suspend option; and wherein injecting the redirect code comprises creating memory space in the application process[[:]], injecting the redirect code into the memory space of the application process[[:]], and ~~use~~ using a create remote thread function to execute the redirect code.

15. (Original) The method, as set forth in claim 10, wherein executing the redirect code comprises:

loading the redirect library of redirect functions;

determining a location of an import table replacement function in the redirect library; and  
executing the import table replacement function.

16. (Original) The method, as set forth in claim 15, wherein loading the redirect library of redirect functions comprises loading a dynamic link library.

17. (Currently Amended) The method, as set forth in claim 15, wherein executing the import table replacement function comprises:

searching an import table of the application process for the ~~set of target~~ at least one function ~~calls~~ call; and

modifying the at least one target-function calls-call to reference one or more redirect functions in the redirect library.

18. (Currently Amended) The method, as set forth in claim 15, wherein executing the import table replacement function comprises:

searching dynamic link libraries of the application process for the ~~set of target~~ at least one function ~~calls-call~~; and

modifying the at least one target-function calls-call to reference one or more redirect functions in the redirect library.

19. (Currently Amended) The method, as set forth in claim 10, further comprising:

receiving user information;

authenticating the user information;

pushing an access policy to the first computing device, wherein the access policy specifies ~~specifying~~ resources accessible by a user associated with the user information ~~to first computing device used by the user.~~

20. (Currently Amended) The method, as set forth in claim 19, wherein intercepting the at least one ~~target-function call~~ comprises intercepting at least one socket function call.

21. (Original) The method, as set forth in claim 19, further comprising executing redirect functions to enable a secured access to a plurality of resources via a firewall.

22. (Currently Amended) A method comprising:

receiving user information;

authenticating the user information;

~~pushing an injector to a first computing device executing an application process; and~~

upon authentication of the user information, intercepting at least one target-function call made by the application process to access at least one of a plurality of secure resources at a remote computing system and executing at least one redirect function in place of the at least one target-function call, the at least one redirect function allows access so as to enable the application process, executing on a first computing device, to access the at least one secure resource at the remote computer system by authorizing access with security in place at the remote computing system.

23. (Currently Amended) The method, as set forth in claim 22, further comprising:

injecting a redirect code into the application process; and  
executing the redirect code in the application process to reference a redirect library of  
redirect functions; ~~and~~  
~~resuming the execution of the application process.~~

24. (Currently Amended) The method, as set forth in claim 23, wherein injecting ~~a~~the redirect  
code ~~further comprises: starting the application process; interrupting the execution of the~~  
~~application process; and~~ injecting the redirect code into a memory space of the application  
process.

25. (Currently Amended) The method, as set forth in claim 23, wherein injecting ~~a~~the redirect  
code ~~further comprises: starting the application process using a debug option; catching an~~  
~~exception thrown by the application process; locating memory space in the application process;~~  
injecting the redirect code into the memory space of the application process; and ~~set~~setting an  
instruction pointer to the redirect code.

26. (Currently Amended) The method, as set forth in claim 23, wherein injecting ~~a~~the redirect  
code ~~further comprises: starting the application process using a suspend option; creating memory~~  
space in the application process; injecting the redirect code into the memory space of the  
application process; and ~~set~~setting an instruction pointer to the redirect code.

27. (Currently Amended) The method, as set forth in claim 23, wherein injecting ~~a~~the redirect  
code ~~further comprises: starting the application process using a suspend option; creating memory~~  
space in the application process; injecting the redirect code into the memory space of the  
application process; and ~~use~~using a create remote thread function to execute the redirect code.

28. (Original) The method, as set forth in claim 23, wherein executing the redirect code  
comprises:

loading the redirect library of redirect functions;  
determining a location of an import table replacement function in the redirect library; and  
executing the import table replacement function.

29. (Original) The method, as set forth in claim 28, wherein loading the redirect library of  
redirect functions comprises loading a dynamic link library.

30. (Currently Amended) The method, as set forth in claim 28, wherein executing the import table replacement function comprises:

searching an import table of the application process for the at least one ~~set of target~~ function ~~calls~~ call; and

modifying the at least one ~~target-function calls~~ call to reference one or more redirect functions in the redirect library.

31. (Currently Amended) The method, as set forth in claim 28, wherein executing the import table replacement function comprises:

searching dynamic link libraries of the application process for the at least one ~~set of target~~ function ~~calls~~ call; and

modifying the at least one ~~target-function calls~~ call to reference one or more redirect functions in the redirect library.

32. (Currently Amended) The method, as set forth in claim 22, wherein intercepting the at least one ~~target~~ function call comprises intercepting at least one socket function call.

33. (Original) The method, as set forth in claim 22, further comprising executing redirect functions to enable a secured access to a plurality of resources via a firewall.